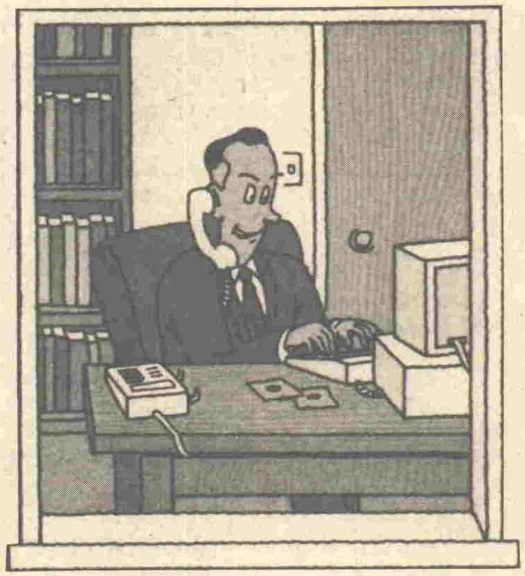
จเร เลิศสุดวิชัย

การใช้งานตัวแปรใน คลิปเปอร์ 5.0



พัฒนาการคลิปเปอร์ 5.0 เพียบพร้อมการใช้งานอย่างนี้จะอดใจไหวหรือ

ในคลิปเปอร์ 5.0 เริ่มปรับปรุงภาษาให้มีลักษณะเป็นภาษาโครงสร้าง (Structure Language) มากขึ้น โดยเฉพาะในเรื่องตัวแปร ซึ่งมีการเปลี่ยนแปลงและเพิ่มเติมในเรื่องของชนิด (Type) ความยาว (Length) ช่วงชีวิตเวลา (life-time) และขอบเขต (Scope) ซึ่งในมุมดีเบสตอนนี้จะแนะนำวิธีการใช้งานตัวแปรของคลิปเปอร์ 5.0

การตั้งชื่อตัวแปร

กฎการตั้งชื่อตัวแปรคลิปเปอร์ก็เหมือนกับในภาษาคอมพิวเตอร์อื่น ๆ ถึงตัวแรกของชื่อจะต้องเป็นตัวอักษร (letter) หรือเครื่องหมายขีดเส้นใต้ (underscore) เท่านั้น และจากนั้นจะเป็นตัวอักษร ตัวขีดเส้นใต้ หรือตัวเลขก็ได้ จำนวนตัวอักษรที่ใช้ในชื่อที่ตั้งนั้นคลิปเปอร์จะจำเพียง 10 ตัวอักษรแรกของชื่อ ดังนั้นไปตั้งชื่อตัวแปรยาวต้องระมัดระวัง เพราะทำให้โปรแกรมทำงานผิดพลาดได้โดยอาจมองเห็นเป็นตัวแปรตัวเดียวกันตัวอย่างเช่น

```
procline1 _ 1
procline1 _ 2
```

ตัวแปรทั้งสองคือตัวแปรตัวเดียวกัน

> กฎ การตั้งชื่อตัวแปรเฉพาะสิบตัวอักษรแรกของชื่อเท่านั้นคลิปเปอร์จะจำ

ชนิดของตัวแปร (Type)

ในคลิปเปอร์ 5.0 ได้เพิ่มชนิดตัวแปรเข้ามาใหม่หลายชนิด นอกจากนั้นของเดิมบางชนิดที่มีอยู่เช่น อะเรย์ก็ได้ปรับปรุงการใช้งานให้มีคุณสมบัติพิเศษเพิ่มเข้ามาอีกซึ่งจะได้พิจารณากันละเอียดอีกครั้งภายหลัง

ตัวแปรในคลิปเปอร์มีหลายชนิดดังนี้

- อะเรย์ (ARRAY)
- บล็อก (BLOCK)
- อักขระ (CHARACTER)
- วันที่ (DATE)
- ลอจิก (LOGICAL)
- เมมโม (MEMO)
- ตัวเลข (NUMERIC จำนวนเต็มหรือจำนวนจริง)
- ออปเจ็กต์ (OBJECT)

ในคลิปเปอร์ยังเหมือนดีเบสอยู่คือชนิดของตัวแปรจะเกิดขึ้นตามค่าข้อมูลที่กำหนดให้มันครั้งแรก

เราสามารถเปลี่ยนค่าของข้อมูลจากอีกชนิดไปเป็นชนิดได้ตามตารางที่ 1 จากตารางสามารถเขียนเป็นคอมพายเลอร์มาโครเอาไว้ใช้งานได้ดังรูปที่ 1

```
* Convert Logical to character
# define ltoc (l) iff (l, "T", "F")
* Convert charecter to logical
# define ctol (C) c $ ("Tt")
```

รูปที่ 1 การเขียนคอมพายเลอร์มาโคร

จาก/เป็น	CHAR	DATE	LOGICAL	NUMERIC
CHAR		ctod (<c>)	<c> $ "Tt"	val (<c>)
DATE	dtoc (<d>)		–	–
LOGICAL	iff (<l>, "T", "F")	–		–
NUMERIC	str (<n>)	–	–	

ตารางที่ 1 การเปลี่ยนค่าชนิดของข้อมูล

สำหรับการเปลี่ยนข้อมูลเป็นอย่างอื่นในคลิปเปอร์มีฟังก์ชันสนับสนุนเอาไว้ให้แล้ว เช่น ฟังก์ชัน STR เปลี่ยนค่าตัวเลขเป็นตัวอักษร ฟังก์ชัน VAL ซึ่งทำงานตรงข้ามกับ STR ส่วนฟังก์ชัน DTOC และ CTOD ทำหน้าที่เปลี่ยนค่าวันที่เป็นตัวอักษร และจากตัวอักษรเป็นค่าวันที่

ความยาว (Lengths)

ในเรื่องของความยาวของตัวแปรจะเกี่ยวกับตัวแปรชนิดตัวอักขระและตัวเลขเท่านั้น ในข้อมูลชนิดตัวอักขระเมื่อทำการกำหนดค่าให้กับตัวแปร ความยาวของตัวแปรจะถูกกำหนดให้กับตัวแปรนั้นทันทีตามความยาวของข้อมูลที่กำหนดให้ เมื่อใช้คำสั่ง GET ในการให้ค่าตัวแปรเราจะไม่สามารถขยายความยาวของตัวแปรออกไปได้อีก ดังนั้นในการกำหนดค่าให้ตัวแปร เพื่อใช้ในคำสั่ง GET จะต้องกำหนดความยาวให้มากที่สุดที่ต้องการใช้งาน

สำหรับในตัวเลขเรื่องของความยาวจะเกี่ยวข้องกับตัวเลข 2 ส่วนคือ ตัวเลขก่อนหน้าจุดทศนิยมและตัวเลขหลังจุดทศนิยม ซึ่งความยาวของตัวเลขทั้งสองนั้นจะมีเฉพาะตอนแสดงผลเท่านั้น จะไม่มีผลต่อการเก็บค่าภายใน เมื่อได้คำสั่ง GET กับตัวแปรตัวเลขคลิปเปอร์จะจัดช่องความยาวให้เติมสำหรับตัวเลขก่อนหน้าจุดทศนิยม 10 ตัวโดยอัตโนมัติ แต่สำหรับช่วงของตัวเลขหลังจุดทศนิยมจะขึ้นอยู่กับการกำหนดค่าเริ่มต้น ตัวอย่างดังรูปที่ 2 ในกรณีนี้ ตัว exch_rate มีจำนวนเลขหลังจุดทศนิยม 3 ตัว แต่ในคำสั่ง GET คลิปเปอร์ก็จะเตรียมช่องสำหรับตัวเลขก่อนจุดทศนิยมให้ 10 ช่องอีก ดังนั้นเมื่อรวมจำนวนช่องทั้งหมดสำหรับในกรณีของตัวแปร exch_rate ก็จะกว้างเป็น 14 ช่อง (รวมจุดทศนิยมด้วย) ถึงแม้ว่าคำสั่ง GET จะเตรียม

ช่วงเอาไว้ให้ 14 ช่วง ก็ตาม การป้อนค่าตัวเลขทศนิยมก็จะใส่ได้เพียง 3 ตัวเท่านั้น ดังนั้นในการใช้คำสั่ง GET กับตัวแปร ตัวเลขการกำหนดค่าเริ่มต้นต้องใส่ค่าทศนิยมหลังจุดให้มีจำนวนตัวเลขเพียงพอกับความต้องการ (ยกเว้นกรณีที่ใช้ Template ด้วยคำสั่ง PICTURE)

ในกรณีที่ทำการเปลี่ยนข้อมูลค่าตัวเลขให้เป็นข้อความด้วยฟังก์ชัน STR ผลลัพธ์ของการเปลี่ยนนั้นคลิปเปอร์จะจัดให้ลงในช่องว่าง 10 ช่อง สำหรับตัวเลขหน้าจุดทศนิยมตามด้วยจุดทศนิยมและตัวเลขหลังทศนิยมจำนวนช่องเท่ากับที่มีอยู่ในขณะนั้น ในกรณีใช้ฟังก์ชัน STR โดยไม่ใส่ความกว้างกำกับ ปัญหาที่อาจปรากฏและจะไม่ค่อยได้รับความสนใจก็คือ ผลลัพธ์ที่เป็นตัวอักษรจะถูกจัดให้ชิดขวา โดยถ้าเกิดจำนวนตัวเลขหน้าจุดทศนิยมมีไม่ถึง 10 ตัว ก็จะปรากฏช่องว่างทางซ้ายแทน ซึ่งถ้าต้องการกำจัดช่องว่างทางซ้ายทิ้งให้ใช้ฟังก์ชัน LTRIM อย่างไรก็ตาม ในการใช้ฟังก์ชัน STR นั้นสามารถจะกำหนดความกว้างของตัวเลขก่อนจุดทศนิยม และตัวเลขหลังทศนิยมได้ด้วยการใส่พารามิเตอร์ตามหลังค่าตัวเลขที่ต้องการเปลี่ยนเป็นตัวอักษร โดยพารามิเตอร์ตัวแรกที่ต่อจากค่าตัวเลขคือ ความกว้างของตัวอักษรทั้งหมดที่จะเกิดขึ้น และถ้าพารามิเตอร์ตัวถัดไปก็คือ ตัวเลขที่บอกถึงจำนวนตัวเลขหลังจุดทศนิยมนั้นเอง ดังตัวอย่างนี้

```
STR (account_id, 6, 2)
```

ในกรณีนี้จะมีช่องว่างสำหรับค่าตัวเลขก่อนทศนิยมในตัวแปร account อยู่เพียง 3 ช่องเท่านั้น อีกสิ่งหนึ่งที่ควรระวังก็คือ ถ้าไม่ได้ใส่พารามิเตอร์ตัวที่ 2 ลงไปการแปลงค่าตัวเลขของคลิปเปอร์จะไม่นำเอาค่าทศนิยมมาใส่ให้อย่างที่ควรจะเป็น นั่นก็คือ ผลลัพธ์จะได้ข้อความที่เป็นจำนวนเต็มแทน และในกรณีกำหนดจำนวนช่วงไว้ไม่พอกับค่าตัวเลขที่จะเปลี่ยน คลิปเปอร์จะใส่ตัวอักษรดอกจันแทนในช่วงของข้อความทั้งหมด

และสุดท้ายที่เหมือนในดีเบสก็คือมีสวิตช์ที่เกี่ยวข้องกับเรื่องการแสดงผลตัวเลข คือ SET DECIMALS TO <n> โดยที่ n คือจำนวนตัวเลขหลังทศนิยมที่ให้ปรากฏในการแสดงผลสวิตช์อีกสวิตช์หนึ่งคือ SET FIXED ON/OFF ซึ่งเป็นสวิตช์ที่จะทำให้เกิดการบังคับการแสดงผลตัวเลขทศนิยมให้ปรากฏตามจำนวนที่สั่งไว้ใน SET DECIMAL ซึ่งตามปกติจะถูก OFF อยู่

สำหรับเรื่องของตัวแปรในคลิปเปอร์ยังมีอีกหลายส่วนที่ยังกล่าวไม่ถึง เช่น เรื่อง ช่วงชีวิตเวลา (Lifetime) และขอบเขต (scope) ซึ่งในตอนหน้าเราจะมาพิจารณากัน แต่ทั้งสองเรื่องนี้จะไปเกี่ยวข้องกับการเขียนโปรแกรมที่เป็นโมดูล (Module) รวมทั้งเรื่องของการสร้างโพรซีเดอร์และฟังก์ชันด้วย พบกันฉบับหน้าครับ

```
exch_rate = 123.456
@ 10,10 SAY "Enter exchage rate" GET exch_rate
READ
```

รูปที่ 2 การเขียนตัวเลขเรื่อง